#26

PATENT

MNFRAME.005A1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant	:	Johnson, et al.	Group Art Unit 2785
Appl. No.	:	08/942,402	
Filed	:	October 1, 1997	
For	:	DIAGNOSTIC AND MANAGING DISTRIBUTED PROCESSOR SYSTEM	
Examiner	:	Norman Wright	

RECEIVED [illegible] GROUP 2100

DECLARATION UNDER 37 C.F.R. § 1.131 TO OVERCOME TAVALLAEI

1. This declaration is to establish the status of the invention in the above-captioned U.S. patent application in the United States on December 31, 1996, which is the effective date of U.S. Patent No. 5,864,653 entitled PCI HOT SPARE CAPABILITY FOR FAILED COMPONENTS, to Tavallaei et al., which was cited by the Examiner against the above-captioned application.

2. We are the named joint inventors of the described subject matter and all claims in the above-referenced regular patent application, filed October 1, 1997.

3. We have read the Office Actions mailed September 15, 1999, (Paper No. 14) and April 26, 2000, (Paper No. 20) regarding the patent application.

4. We reduced to practice the invention described and claimed in the pending application by at least December 19, 1996, as evidenced by the following events:

   a. By at least December 19, 1996, NetFRAME (the assignee of the subject application) was manufacturing and selling a fully functional product (the NF9000 family of network servers) that reduced to practice the claimed subject matter. The commercial product was described as being commercially available in a document entitled "Novell IntranetWare supports hot pluggable PCI from NetFRAME," which was published on December 19, 1996, as evidenced by the document date. A copy of page 1 is attached as **Exhibit A.**

b. A document entitled "Raptor Wire Service Architecture, Version 1.3" ("RWSA"), dated October 3, 1996, shows the invention, in its entirety, as claimed in independent Claims 1, 11, 19, and 20 and as reduced to practice in the product sold by NetFRAME. Copies of the cover and page 1 of RWSA are attached as **Exhibit B.** For reference, Claim 1 (as amended) of the pending patent application recites a "computer monitoring and diagnostic system comprising:

> a computer, having a computing device and a housing;
> a plurality of sensors, located within the computer, capable of sensing conditions within the computer; and
> a microcontroller network, located within the computer, said network comprising a plurality of interconnected microcontrollers, connected to the sensors and the computer, wherein the microcontroller network processes requests for conditions from the computer and responsively provides sensed conditions to the computer.

Page 1 of RWSA depicts a "Wire Service Hardware Diagram" (a more readable version of the same diagram is presented as Fig. 5A and Fig. 5B of the patent application). RWSA shows a computer (e.g., the ISA Bus to communicate with the CPU, a PCI card, and dual CPUs on a Motherboard), a plurality of sensors, located within the computer, capable of sensing the conditions of the computer (e.g., Temperature Detector on Backplane, Temperature Detector on Motherboard, CPU Thermal Fault Detector, Fan Speed Detector, etc.), and a microcontroller network, located within the computer, comprising a plurality of microcontrollers (e.g., Chassis Controller on Back Plane board, CPU A controller on Motherboard, etc.) communicating on a microcontroller bus, i.e., the *Wire Service Bus*.

Page 1 of RWSA also shows that the microcontroller network is connected to the sensors (e.g., Chassis Controller is connected to Temperature Detector on Backplane and Temperature Detector on Motherboard; CPU B Controller is connected to CPU Thermal Fault Detector) and the computer (e.g., the microcontroller network connects to the CPU through the System Interface and the ISA Bus). RWSA shows motherboard, backplane, and environmentally related ambient temperature detectors connected to the microcontroller bus through the Chassis Controller.

Additionally, page 1 of RWSA shows that the microcontroller network depicted is capable of processing requests for conditions from the computer and responsively providing sensed conditions to the computer via the ISA Bus. For example, Fan Speed Detection and Fan Speed control signals are accessible by the microcontroller bus through CPU A Controller.

Thus, **Exhibit B** depicts all of the claimed limitations of Claim 1. Furthermore, The invention was intended to monitor and diagnose the environmental conditions of the computer. The claimed subject matter, as depicted in RWSA, and as incorporated into the commercial product sold by NetFRAME, worked for its intended purpose.

c. By at least January 1996, we had conceived of using a network of microcontrollers as the monitoring and control hardware of the subject invention. This is shown by a document entitled "Raptor Wire Service Architecture, Version 1.0" ("Wire Architecture"), written at least as early as January 23, 1996, as evidenced by the document date. A copy of the cover page and pages 6-8 and 13-25 of Wire Architecture is attached as **Exhibit C**.

Pages 6-8 of Wire Architecture describe a log data type used in requesting and recording sensed conditions of the computer. The event data type for alerting the source of the request of sensed conditions in the Wire Service (microcontroller network) includes those relating to environmental conditions such as Fan Status Change.

Wire Architecture also discloses a table of Wire Service Network Physical Connections at pages 13-19. This table describes one embodiment of the physical signal connections to all the Wire Service processors for conditions including fan tachometer (speed measurement), temperature bus from probes at different points, and fan fault indicators. The data on the microcontroller network can be monitored by the source of a request for condition data. This illustrates the limitation of communicating the sensed conditions (from the probes) from the microcontroller network to the source of the request.

5. I, Karl S. Johnson, am listed as an inventor on a provisional Patent Application No. 60/046,397, filed May 13, 1997, which is a priority application for the subject application.

6. All acts leading to the reduction of practice were performed in the United States.

We declare that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful, false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful, false statements may jeopardize the validity of the application or any patent resulting therefrom.

Dated: August 14, 2000   By: [illegible]
Karl Johnson

Dated: ____________   By: ____________
Walter Wallach

Dated: ____________   By: ____________
Carlton Amdahl

Dated: ____________   By: ____________
Ken Nguyen

S:\DOCS\JFK\JFK-1022.DOC
062700

Appl. No. : 08/942,402
Filed : October 1, 1997

monitored by the source of a request for condition data. This illustrates the limitation of communicating the sensed conditions (from the probes) from the microcontroller network to the source of the request.

5. I, Karl S. Johnson, am listed as an inventor on a provisional Patent Application No. 60/046,397, filed May 13, 1997, which is a priority application for the subject application.

6. All acts leading to the reduction of practice were performed in the United States.

We declare that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful, false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful, false statements may jeopardize the validity of the application or any patent resulting therefrom.

Dated: ____________________ By: ____________________
Karl Johnson

Dated: ____________________ By: ____________________
Walter Wallach

Dated: 8-11-00 By: [illegible]
Carlton Amdahl

Dated: ____________________ By: ____________________
Ken Nguyen

S:\DOCS\JFK\JFK-1022.DOC
062700

Appl. No. : 08/942,402
Filed : October 1, 1997

monitored by the source of a request for condition data. This illustrates the limitation of communicating the sensed conditions (from the probes) from the microcontroller network to the source of the request.

5. I, Karl S. Johnson, am listed as an inventor on a provisional Patent Application No. 60/046,397, filed May 13, 1997, which is a priority application for the subject application.

6. All acts leading to the reduction of practice were performed in the United States.

We declare that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful, false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful, false statements may jeopardize the validity of the application or any patent resulting therefrom.

Dated: ____________	By: ____________ Karl Johnson
Dated: 8/24/00	By: [illegible] Walter Wallach
Dated: ____________	By: ____________ Carlton Amdahl
Dated: ____________	By: ____________ Ken Nguyen

S:\DOCS\JFK\JFK-1022.DOC
062700

EXHIBIT A

03393902/9
DIALOG(R)File 636:Gale Group Newsletter DB(TM)

03393902 Supplier Number: 46983928 (THIS IS THE FULLTEXT)
NOVELL: Novell IntranetWare supports hot pluggable PCI from NetFRAME
M2 Presswire, pN/A
Dec 19, 1996
Language: English Record Type: Fulltext
Document Type: Newswire; Trade
Word Count: 495
TEXT:
M2 PRESSWIRE-19 December 1996-NOVELL: Novell IntranetWare supports hot pluggable PCI from NetFRAME (C)1994-96 M2 COMMUNICATIONS LTD RDATE:181296 * IntranetWare customers can add and swap PCI cards in on-line systems with minimal server downtime Novell, Inc. today announced that customers using IntranetWare, Novell's full-service Internet/intranet access platform, can take advantage of both Hot Add and Hot Swap PCI with today's NetFRAME enterprise-class network servers. The companies will continue to work closely in the future to ensure that the recently proposed PCI Hot Plug standard will deliver the level of functionality that IntranetWare and NetFRAME customers depend on.

"Hot Pluggable PCI is a key technology for continual Internet and intranet availability," said William Donahoo, senior director of product marketing at Novell. "With today's requirement for 24-hour information access, server downtime resulting from server component failure, system maintenance or hardware expansion is unacceptable. Supporting this new technology brings a new level of flexibility and fault tolerance that helps customers build business-critical intranets."

Hot Pluggable PCI technology from NetFRAME, introduced October, 1996, enables IntranetWare customers to add and swap industry standard PCI boards and device drivers, while users remain on-line greatly reducing server downtime and service disruption. The technology supports PCI-based SCSI, Ethernet, FDDI and Token Ring interface cards and device drivers. System administrators can use this functionality to both repair and expand server storage and network connectivity without having to bring down either IntranetWare or the server.

"Novell is a leader in the network operating system market," said Steve Huey, vice president of marketing at NetFRAME. "We believe Novell is well positioned to shape the future of continuous intranet computing as organizations evolve their LANs into intranets. By shipping Hot Pluggable PCI technology today, NetFRAME makes it possible for IntranetWare users to deploy continuously available server environments."

By combining IntranetWare's unique ability to load and unload device drivers without downing the server with NetFRAME's Hot Pluggable PCI technology, system administrators can add new PCI devices to a server with no user downtime. For example, if a server's network adapter fails, it can be replaced without requiring an administrator to take IntranetWare off-line or re-booting the server. When a component is replaced, the card and driver are automatically identified and configured, and the card is instantly made available as a system resource.

Founded in 1983, Novell (NASDAQ: NOVL) is the world's leading provider of network software. The company offers a wide range of network solutions for distributed network, Internet, intranet and small-business markets. Novell education and technical support programs are the most comprehensive in the network computing industry. Information about Novell's complete range of products and services can be accessed on the World Wide Web at http://www.novell.com.

Novell is a registered trademark and IntranetWare is a trademark of Novell, Inc. All other registered trademarks and trademarks are the property of their respective holders.

M2 COMMUNICATIONS DISCLAIMS ALL LIABILITY FOR INFORMATION PROVIDED WITHIN M2 PRESSWIRE. DATA SUPPLIED BY NAMED PARTY/PARTIES.

EXHIBIT B

Raptor Wire Service Architecture

Version 1.3

October 3, 1996

Prepared for
NetFrame Raptor Implementation Group

by
Karl Johnson (KJ)

Introduction

"Wire Service" is the code name for the Raptor project system control, diagnostic and maintenance bus (formerly known as the CDM bus). Raptor is a completely "fly by wire" system - no switch, indicator or other control is directly connected to the function it monitors or controls. Instead, all the control and monitoring connections are made by the network of processors that comprise the "Wire Service" for the system. The processors are Microchip PIC processors and the network is a 400 kbps I^{2}C serial bus. A limited understanding of I^{2}C protocol is a prerequisite for understanding Wire Service protocols (See "The I^{2}C-bus and how to use it" - Philips Semiconductor, Jan 1992). Control on this bus is distributed, each processor can be either a master or a slave and can control resources on itself or any other processor on the bus.

Wire Service Hardware Block Diagram

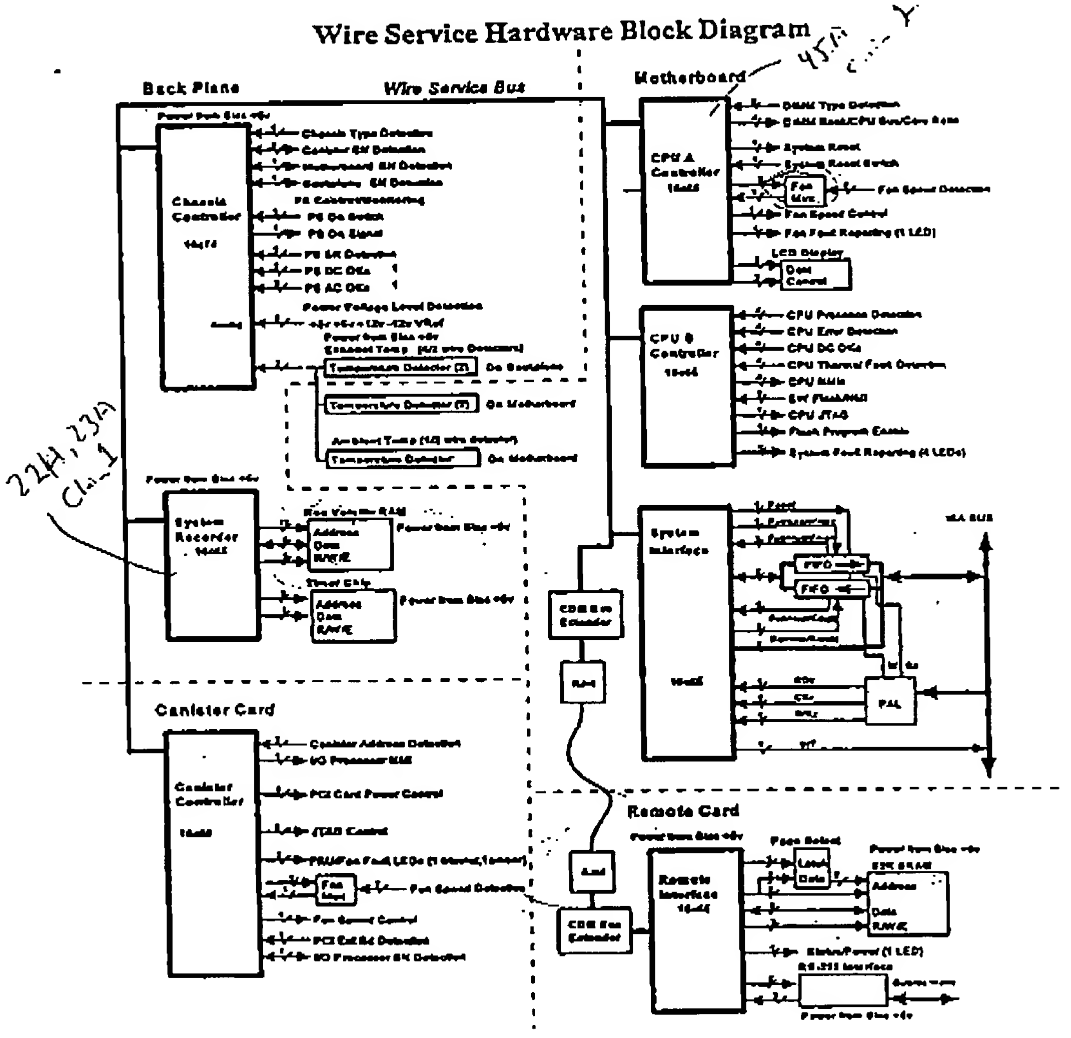

EXHIBIT C

Raptor Wire Service Architecture

Version 1.0

1/23/96

Prepared for
Raptor Implementation Group

by
Karl Johnson (KJ)

Write Byte Array Message

Request

Slave Address	Type/RW	Start Addr MSB	Start Addr LSB	Length N	Array Data 1	...	Array Data N
Check Byte							

Response

Slave Address	Length 0	Status 0/Success	Check Byte

Log Type

(cl. 23

The Log data type is to be used for logging byte strings in circular log buffer. It is used to record system events in the NVRAM system log.

Read Log Message

Request

Slave Address	Type/RW	Log Addr MSB	Log Addr LSB	Request 255	Check Byte

Response

Slave Address	Length N+7	Log Time MSB	Log Time	Log Time	Log Time LSB	Log Addr MSB	Log Addr LSB
Log Data Byte 0	...	Log Data Byte N	Status 0/Success	Check Byte			

Write Log Message

Request

Slave Address	Type/RW	N/A	N/A	Length N	Log Data Byte 0	...	Log Data Byte N
Check Byte							

Response

Slave Address	Length 0	Status 0/Success	Check Byte

The addressing of log entries has some special characteristics.
1) Reading address 65565 (0xffff) is special - It represents the address of the latest entry in the log.
2) Reading address 65564 (0xfffe) is also special - It represents the address of the earliest available entry.

3) The address of real log entries wraps at 65519 (0xffef). The next sequential entry after 65519 is 0.
4) The address of is ignored on write and the next available entry is written.
5) To read the entire log in forward time order, read entry at address 65564. This returns the first log entry along with its actual log address. Increment that address by one and read that entry. Repeat the last step until status indicates failure.
6) To read the entire log in reverse time order, read entry at address 65565. This returns the last log entry along with its actual log address. Decrement that address by one and read that entry. Repeat the last step until status indicates failure.
7)To keep a complete external copy of the log, first read the entire log in forward time order and remember the last valid entry. Then periodically read forward from the remembered last valid entry to the end and add that to the external copy.

Event Type

The event data type is to be used for alerting external interfaces of events in the Wire Service network. Event memory is organized as a queue. The queue will probably be quite small (< 20 Events). Writing an event places the event ID at the next available entry, unless the last queue entry would be written by this event. In that case, the last queue entry is a Queue Overflow Event and the write fails. This allows the external interface to realize that events were lost and it should scan for any changes in data.
Reading the event type returns requested number of events in the queue or the entire queue which ever is less and removes them from the queue.

Read Event Message
Request

Slave Address	Type/RW	N/A	N/A	Request 1-255	Check Byte

Response

Slave Address	Length N	Event ID 1	...	Event ID N	Status 0/Success	Check Byte

Write Event Message

Request

Slave Address	Type/RW	N/A	N/A	Length 1	Event ID	Check Byte

Response

Slave Address	Length 0	Status 0/Success	Check Byte

Possible Event Types:
CPU Status Change
Power Status Change
Canister Status Change
Fan Status Change

Screen Type

The screen data type is to be used for communication of character mode screen information from the system BIOS to remote management Interface.

Read Screen Message

Request

Slave Address	Type/RW	S Addr MSB	S Addr LSB	Request 1-255	Check Byte

Response

Slave Address	Length N	Screen Data 1	...	Screen Data N	Status 0/Success	Check Byte

Write Screen Message

Request

Slave Address	Type/RW	S Addr MSB	S Addr LSB	Length N	Screen Data 1	...	Screen Data n
Check Byte							

Response

Slave Address	Length 0	Status 0/Success	Check Byte

Wire Service Network Physical Connections

The following table describe all of the physical signal connections to all of the Wire Service processors. The names for the connections will be related to network accessible memory data in the section which follows called "Wire Service Network Memory Map".

Note: All signal types and definitions are from the viewpoint of the individual Wire Service PIC processor (e.g. Input means input to PIC processor)

Wire Service System Bus Interface (System Type ID: S0) Processor ID 10

Pin	Type	Name	Function	Notes
RA0	I	S0_FIFO_IEFZ	In FIFO (ISA Writes) Empty Flag (Active Low)	Status Flags for both ISA bus FIFO's. Need to monitor for incoming message overruns. Also must assure that output FIFO is empty before loading output message.
RA1	I	S0_FIFO_IHFZ	In FIFO (ISA Writes) Half-full Flag (Active Low)	
RA2	I	S0_FIFO_IFFZ	In FIFO (ISA Writes) Full Flag (Active Low)	
RA3	I	S0_FIFO_OEFZ	Out FIFO (ISA Reads) Empty Flag (Active Low)	
RA4	I	S0_FIFO_OHFZ	Out FIFO (ISA Reads) Half-full Flag (Active Low)	
RA5	I	S0_FIFO_OFFZ	Out FIFO (ISA Reads) Full Flag (Active Low)	
RB0	I/O	S0_FIFO_D0	ISA FIFOs Data bus Bit 0	This is an 8 bit bi-directional port for the ISA FIFO data bus. These bits should drive the bus before asserting FIFO Write and can be read after tri-stating and asserting FIFO Read
RB1	I/O	S0_FIFO_D1	ISA FIFOs Data bus Bit 1	
RB2	I/O	S0_FIFO_D2	ISA FIFOs Data bus Bit 2	
RB3	I/O	S0_FIFO_D3	ISA FIFOs Data bus Bit 3	
RB4	I/O	S0_FIFO_D4	ISA FIFOs Data bus Bit 4	
RB5	I/O	S0_FIFO_D5	ISA FIFOs Data bus Bit 5	
RB6	I/O	S0_FIFO_D6	ISA FIFOs Data bus Bit 6	
RB7	I/O	S0_FIFO_D7	ISA FIFOs Data bus Bit 7	
RC0	O	S0_FIFO_RZ	FIFO (ISA Writes) Read (Assert Low)	Assert to read the In FIFO
RC1	O	S0_FIFO_WZ	PIC to ISA FIFO (ISA Reads) Write (Assert Low)	Assert to write the Out FIFO
RC2	O	S0_ISA_INT	PIC to ISA Interrupt Request (Assert 7)	Level Interrupt to ISA bus
RC3	I/O	S0_I2C_DATA	Wire Service Bus Clock (I2C)	Only used for I2C
RC4	I/O	S0_I2C_CLK	Wire Service Bus Data (I2C)	
RC5	O	S0_FIFO_RSTZ	In and Out FIFOs Reset (Assert Low)	Resets both FIFOs
RC6	O	S0_FIFO_IRTZ	In FIFO (ISA Writes) Retransmit (Assert Low)	Resets the Read pointer to 0
RC7	O	S0_FIFO_ORTZ	Out FIFO (ISA Reads) Retransmit (Assert Low)	Resets the Read pointer to 0
RD0	I/O	S0_CSR_D0	ISA External Data bus Bit 0 (Slave parallel port)	The slave parallel port is used as a bidirectional control and status register on the ISA bus.
RD1	I/O	S0_CSR_D1	ISA External Data bus Bit 1 (Slave parallel port)	
RD2	I/O	S0_CSR_D2	ISA External Data bus Bit 2 (Slave parallel port)	
RD3	I/O	S0_CSR_D3	ISA External Data bus Bit 3 (Slave parallel port)	
RD4	I/O	S0_CSR_D4	ISA External Data bus Bit 4 (Slave parallel port)	
RD5	I/O	S0_CSR_D5	ISA External Data bus Bit 5 (Slave parallel port)	
RD6	I/O	S0_CSR_D6	ISA External Data bus Bit 6 (Slave parallel port)	
RD7	I/O	S0_CSR_D7	ISA External Data bus Bit 7 (Slave parallel port)	
RE0	I	S0_CSR_RZ	ISA Read Slave Parallel Port (Assert Low)	Not directly manipulated after setting port O/E to act as slave parallel port
RE1	I	S0_CSR_WZ	ISA Write Slave Parallel Port (Assert Low)	
RE2	I	S0_CSR_SZ	ISA Slave Parallel Port Select (Assert Low)	

Wire Service System Monitor A (System Type ID 51) Processor ID 3

Pin	Type	Name	Function	Notes
RA0	O	S1_FAN_HI	System Board fan speed to high (Assert Hi)	Assert on any SB fan failure
RA1	O	S1_SBFAN_LED	System Board fan fault LED	Assert on any SB fan failure
RA2	O	S1_BC_DS0	Bus/Core Speed Ratio and DIMM Select Mux Bit 0	During system reset these bits select bus/core speed ratio for all processors. Otherwise they select which DIMM presents its type on DIMM type port.
RA3	O	S1_BC_DS1	Bus/Core Speed Ratio and DIMM Select Mux Bit 1	
RA4	O	S1_BC_DS2	Bus/Core Speed Ratio and DIMM Select Mux Bit 2	
RA5	O	S1_BC_DS3	Bus/Core Speed Ratio and DIMM Select Mux Bit 3	
RB0	I/O	S1_LCD_D0	LCD Controller Data Bus bit 0	These lines make up the 8 bit data bus to the LCD display
RB1	I/O	S1_LCD_D1	LCD Controller Data Bus bit 1	
RB2	I/O	S1_LCD_D2	LCD Controller Data Bus bit 2	
RB3	I/O	S1_LCD_D3	LCD Controller Data Bus bit 3	
RB4	I/O	S1_LCD_D4	LCD Controller Data Bus bit 4	
RB5	I/O	S1_LCD_D5	LCD Controller Data Bus bit 5	
RB6	I/O	S1_LCD_D6	LCD Controller Data Bus bit 6	
RB7	I/O	S1_LCD_D7	LCD Controller Data Bus bit 7	
RC0	I	S1_FAN_TP	Tachometer pulse input from selected fan	Generally routed to counter
RC1	O?	S1_OK_TO_RUN	Drives SYS_PWRGOOD signal	System starts on 0->1 transition
RC2	I	S1_RESET_SW	Undebounced input from System Reset switch	
RC3	I/O	S1_I2C_DATA	Wire Service Bus Clock (I2C)	Only used for I2C
RC4	I/O	S1_I2C_CLK	Wire Service Bus Data (I2C)	
RC5	O	S1_FAN_SEL0	Fan Tachometer Multiplexer Select Bit 0	Used to select which fan tachometer pulse output is gated to S1_FAN_TP
RC6	O	S1_FAN_SEL1	Fan Tachometer Multiplexer Select Bit 1	
RC7	O	S1_FAN_SEL2	Fan Tachometer Multiplexer Select Bit 2	
RD0	I	S1_DIMM_D0	DIMM Type port bit 0	These lines make up an 8 bit port which on which the DIMM module in the slot selected by S1_BC_DS0..3 presents its type data if any. If no DIMM is present in the slot selected the DIMM type bits are all 1's.
RD1	I	S1_DIMM_D1	DIMM Type port bit 1	
RD2	I	S1_DIMM_D2	DIMM Type port bit 2	
RD3	I	S1_DIMM_D3	DIMM Type port bit 3	
RD4	I	S1_DIMM_D4	DIMM Type port bit 4	
RD5	I	S1_DIMM_D5	DIMM Type port bit 5	
RD6	I	S1_DIMM_D6	DIMM Type port bit 6	
RD7	I	S1_DIMM_D7	DIMM Type port bit 7	
RE0	O	S1_LCD_RS	LCD Controller Register Select	See LCD Controller data sheet for details of operation of these signals
RE1	O	S1_LCD_ENA	LCD Controller Register Enable	
RE2	O	S1_LCD_RW	LCD Controller Register Read/Write	

Wire Service System Monitor B (System Type ID S2) Processor ID 4

Pin	Type	Name	Function	Notes
RA0	O	S2_FLASH_LED	CPU Display the Enable/Disable state of the BIOS Flash ROM	Should track S2_FLASH_ENA
RA1	O	S2_SBFLT_LED0	System Board FRU LED Pin 0 (bicolor LED)	Drive in different combinations for OFF, AMBER, GREEN
RA2	O	S2_SBFLT_LED1	System Board FRU LED Pin 1 (bicolor LED)	
RA3	O	S2_OVRTMP_LED	Over Temperature LED	
RA4	I	S2_TEMP_CPU4	Thermal Fault - CPU 4	Indicator that CPU has exceeded temperature limit and faulted
RA5	I	S2_TEMP_CPU3	Thermal Fault - CPU 3	
RB0	O	S2_SB_JTAG	Enable System Board JTAG Chain TMS	
RB1	O	S2_FLASH_WE	System BIOS FLASH Write Enable	
RB2	I	S2_FLASH_SW	System BIOS FLASH Write Enable Switch (undebounced)	
RB3	I	S2_NMI_SW	System Non-Maskable Interrupt (NMI) Switch (undebounced)	
RB4	I	S2_POK_CPU1	Power Good signal from CPU 1	Indicator that power regulator for CPU is operating correctly. Only valid if corresponding CPU is present (S2_PRES_CPUx)
RB5	I	S2_POK_CPU2	Power Good signal from CPU 2	
RB6	I	S2_POK_CPU3	Power Good signal from CPU 3	
RB7	I	S2_POK_CPU4	Power Good signal from CPU 4	
RC0	x		Unused	
RC1	x		Unused	
RC2	O	S2_NMI_CPU4	NMI Request for CPU 4	Toggle to cause NMI to CPU
RC3	I/O	S2_I2C_CLK	Wire Service Bus Data (I2C)	Only used for I2C
RC4	I/O	S2_I2C_CLK	Wire Service Bus Data (I2C)	
RC5	O	S2_NMI_CPU3	NMI Request for CPU 3	See S2_NMI_CPU4 above
RC6	O	S2_NMI_CPU2	NMI Request for CPU 2	
RC7	O	S2_NMI_CPU1	NMI Request for CPU 1	
RD0	I	S2_PRES_CPU1	Presence detection bit - CPU 1	Asserted when a processor inserted in the system board
RD1	I	S2_PRES_CPU2	Presence detection bit - CPU 2	
RD2	I	S2_PRES_CPU3	Presence detection bit - CPU 3	
RD3	I	S2_PRES_CPU4	Presence detection bit - CPU 4	
RD4	I	S2_ERROR_CPU1	Processor Fault bit - CPU 1	Processor either failed BIST on startup or later other fault. Only valid if corresponding CPU is present (S2_PRES_CPUx)
RD5	I	S2_ERROR_CPU2	Processor Fault bit - CPU 2	
RD6	I	S2_ERROR_CPU3	Processor Fault bit - CPU 3	
RD7	I	S2_ERROR_CPU4	Processor Fault bit - CPU 4	
RE0	O	S2_SYSFLT_LED	System Fault summary LED	
RE1	I	S2_TEMP_CPU2	Thermal Fault - CPU 2	See S2_TEMP_CPU4 above
RE2	I	S2_TEMP_CPU1	Thermal Fault - CPU 1	

Wire Service System Recorder (System Type ID S3) Processor ID 1

Pin	Type	Name	Function	Notes
RA0	O	S3_NVRAM_A8	NVRAM Address Bit 8	NVRAM Address Bus Bits 8-13
RA1	O	S3_NVRAM_A9	NVRAM Address Bit 9	
RA2	O	S3_NVRAM_A10	NVRAM Address Bit 10	
RA3	O	S3_NVRAM_A11	NVRAM Address Bit 11	
RA4	O	S3_NVRAM_A12	NVRAM Address Bit 12	
RA5	O	S3_NVRAM_A13	NVRAM Address Bit 13	
RB0	I/O	S3_NVRAM_D0	NVRAM Data Bit 0	NVRAM 8 Bit Data Bus
RB1	I/O	S3_NVRAM_D1	NVRAM Data Bit 1	
RB2	I/O	S3_NVRAM_D2	NVRAM Data Bit 2	
RB3	I/O	S3_NVRAM_D3	NVRAM Data Bit 3	
RB4	I/O	S3_NVRAM_D4	NVRAM Data Bit 4	
RB5	I/O	S3_NVRAM_D5	NVRAM Data Bit 5	
RB6	I/O	S3_NVRAM_D6	NVRAM Data Bit 6	
RB7	I/O	S3_NVRAM_D7	NVRAM Data Bit 7	
RC0	O	S3_NVRAM_CSZ	NVRAM Chip Select (Negative Logic)	Control signals for NVRAM - See Dallas DS1245 data sheet
RC1	O	S3_NVRAM_OEZ	NVRAM Output Enable (Negative Logic)	
RC2	O	S3_NVRAM_WEZ	NVRAM Write Enable (Negative Logic)	
RC3	I/O	S3_I2C_CLK	Wire Service Bus Data (I2C)	Only used for I2C
RC4	I/O	S3_I2C_CLK	Wire Service Bus Data (I2C)	
RC5	O	S3_NVRAM_A14	NVRAM Address Bit 14	NVRAM Address Bus Bits 14-16
RC6	O	S3_NVRAM_A15	NVRAM Address Bit 15	
RC7	O	S3_NVRAM_A16	NVRAM Address Bit 16	
RD0	O	S3_NVRAM_A0	NVRAM Address Bit 0	NVRAM Address Bus Bits 0-7
RD1	O	S3_NVRAM_A1	NVRAM Address Bit 1	
RD2	O	S3_NVRAM_A2	NVRAM Address Bit 2	
RD3	O	S3_NVRAM_A3	NVRAM Address Bit 3	
RD4	O	S3_NVRAM_A4	NVRAM Address Bit 4	
RD5	O	S3_NVRAM_A5	NVRAM Address Bit 5	
RD6	O	S3_NVRAM_A6	NVRAM Address Bit 6	
RD7	O	S3_NVRAM_A7	NVRAM Address Bit 7	
RE0	O	S3_RTC_CLK	Real Time Clock - Data Clock	See Dallas DS1603 data sheet
RE1	I	S3_RTC_DATA	Real Time Clock - Serial Data	
RE2	O	S3_RTC_RSTZ	Real Time Clock - Protocol Reset (Negative Logic)	

Wire Service Backplane (System Type ID S4) Processor ID 2

Pin	Type	Name	Function	Notes
RA0	A	S4_VOLTS_P5V	Analog measure of system +5 volt main supply	Use D/A converter to read voltages as 0-255. Calibration constants are determined externally
RA1	A	S4_VOLTS_P3V	Analog measure of system +3.3 volt main supply	
RA2	A	S4_VOLTS_P12V	Analog measure of system +12 volt main supply	
RA3	A	S4_VREF	Voltage Reference for A/D converter	Unused
RA4	x			
RA5	A	S4_VOLTS_N12V	Analog measure of system -12 volt main supply	See S4_VOLTS_P5V
RB0	I/O	S4_PSN_CAN1	Presence and Serial Number I/O for Canister 1	These are all lines to one wire serial data EPROMS. See Dallas DS250x data sheet for programming information
RB1	I/O	S4_PSN_CAN2	Presence and Serial Number I/O for Canister 2	
RB2	I/O	S4_PSN_CAN3	Presence and Serial Number I/O for Canister 3	
RB3	I/O	S4_PSN_CAN4	Presence and Serial Number I/O for Canister 4	
RB4	I/O	S4_PSN_CAN5	Presence and Serial Number I/O for Canister 5	
RB5	I/O	S4_PSN_CAN6	Presence and Serial Number I/O for Canister 6	
RB6	I/O	S4_PSN_CAN7	Presence and Serial Number I/O for Canister 7	
RB7	I/O	S4_PSN_CAN8	Presence and Serial Number I/O for Canister 8	
RC0	x			
RC1	I	S4_ACOK_PS3	A/C Input OK to Power Supply 3	Should check only if PSN for power supply indicates presence.
RC2	I	S4_ACOK_PS2	A/C Input OK to Power Supply 2	
RC3	I/O	S4_I2C_CLK	Wire Service Bus Data (I2C)	Only used for I2C
RC4	I/O	S4_I2C_CLK	Wire Service Bus Data (I2C)	
RC5	I	S4_ACOK_PS1	A/C Input OK to Power Supply 1	See S4_ACOK_PS3
RC6	O	S4_POWER_ON	Enable main output from power supplies	
RC7	I	S4_POWER_SW	Power On/Off switch (undebounced)	
RD0	I/O	S4_PSN_PS1	Presence and Serial Number for Power Supply 1	These are all lines to one wire serial data EPROMS. See Dallas DS250x data sheet
RD1	I/O	S4_PSN_PS2	Presence and Serial Number for Power Supply 2	
RD2	I/O	S4_PSN_PS3	Presence and Serial Number for Power Supply 3	
RD3	I/O	S4_PSN_BP	Presence and Serial Number for Backplane	Dallas DS250x also
RD4	I/O	S4_PSN_SB	Presence and Serial Number for System Board	Dallas DS250x also
RD5	I	S4_BP_TYPE	Backplane Type (0: Small 1: Large)	
RD6	I/O	S4_TEMP_SCL	Temperature Bus Clock	I2C local bus for temperature probes at different system points
RD7	I/O	S4_TEMP_SDA	Temperature Bus Serial Data	
RE0	I	S4_DCOK_PS3	D/C Output OK from Power Supply 3	Should check only if PSN for power supply indicates presence.
RE1	I	S4_DCOK_PS2	D/C Output OK from Power Supply 2	
RE2	I	S4_DCOK_PS1	D/C Output OK from Power Supply 1	

Wire Service Canister (System Type ID S5) Processor ID 2x where x is the slot ID

Pin	Type	Name	Function	Notes
RA0	O	S5_P12V_ENA	Turns on +/- 12 volt to all PCI slots	Used to sequence power to PCI cards.
RA1	O	S5_P5V_ENA4	Turns on +5 volts to PCI slot 4	
RA2	O	S5_P5V_ENA3	Turns on +5 volts to PCI slot 3	
RA3	O	S5_P5V_ENA2	Turns on +5 volts to PCI slot 2	
RA4	O	S5_P5V_ENA1	Turns on +5 volts to PCI slot 1	
RA5	?			
RB0	I	S5_CAN_A0	Canister Address bit 0	Determine the Wire Service bus address of this canister
RB1	I	S5_CAN_A1	Canister Address bit 1	
RB2	I	S5_CAN_A2	Canister Address bit 2	
RB3	I	S5_PRSNT_S5	Special Slot 5 (IOP/PCI jumper) present	Indicates something is in slot 5
RB4	I/O	S5_PSN_S5	Present Serial Number for special slot 5	From DS 250x in slot 5 card (if IOP)
RB5	x			
RB6	x			
RB7	x			
RC0	I	S5_FAN_TP	Tachometer pulse input from selected fan	Generally routed to counter
RC1	O	S5_FAN_SEL0	Fan Tachometer Multiplexer Select Bit 0	Select which fan to monitor tach.
RC2	x			
RC3	I/O	S5_I2C_CLK	Wire Service Bus Data (I2C)	Only used for I2C
RC4	I/O	S5_I2C_CLK	Wire Service Bus Data (I2C)	
RC5	O	S5_CANFAN_LED	Canister fan fault LED	Assert on any Canister fan failure
RC6	O	S5_CANFLT_LED0	Canister FRU LED Pin 0 (bicolor LED)	Drive in different combinations for OFF, AMBER, GREEN
RC7	O	S5_CANFLT_LED1	Canister FRU LED Pin 1 (bicolor LED)	
RD0	I	S5_PRSNT_S1A	PCI card present in Slot 1 (A pin)	PCI slots have 2 presence pins - see PCI spec for usage and meaning.
RD1	I	S5_PRSNT_S1B	PCI card present in Slot 1 (B pin)	
RD2	I	S5_PRSNT_S2A	PCI card present in Slot 2 (A pin)	
RD3	I	S5_PRSNT_S2B	PCI card present in Slot 2 (B pin)	
RD4	I	S5_PRSNT_S3A	PCI card present in Slot 3 (A pin)	
RD5	I	S5_PRSNT_S3B	PCI card present in Slot 3 (B pin)	
RD6	I	S5_PRSNT_S4A	PCI card present in Slot 4 (A pin)	
RD7	I	S5_PRSNT_S4B	PCI card present in Slot 4 (B pin)	
RE0	O	S5_CAN_JTAG	Enable Canister-Board JTAG Chain TMS	Required to select the JTAG chain
RE1	O	S5_NMI_S5	NMI card is special slot 5 (IOP)	Toggle to NMI IOP in slot 5
RE2	O	S2_FAN_HI	Canister fan speed to high (Assert HI)	Assert on any Canister fan failure

Wire Service Remote Interface (System Type ID S6) Processor ID 11

Pin	Type	Name	Function	Notes
RA0	I/O	S6_PSN_RI	Serial Number Information for Remote Interface	
RA1	x			
RA2	x			
RA3	x			
RA4	x			
RA5	x			
RB0	O	S6_MODEM_DTR	Modem Signal (Data Terminal Ready)	
RB1	I	S6_MODEM_DSR	Modem Signal (Data Set Ready)	
RB2	I	S6_MODEM_CD	Modem Signal (Carrier Detect)	
RB3	I	S6_MODEM_RI	Modem Signal (Ring Indicate)	
RB4	O	S6_MODEM_RTS	Modem Signal (Request To Send)	
RB5	I	S6_MODEM_CTS	Modem Signal (Clear To Send)	
RB6	x			
RB7	x			
RC0	x			
RC1	x			
RC2	x			
RC3	I/O	S5_I2C_CLK	Wire Service Bus Data (I2C)	Only used for I2C
RC4	I/O	S5_I2C_CLK	Wire Service Bus Data (I2C)	
RC5	x			
RC6	O	S6_MODEM_TXD	Modem Signal (Transmit Data)	Controlled by chip serial interface
RC7	I	S6_MODEM_RXD	Modem Signal (Receive Data)	
RD0	x			
RD1	x			
RD2	x			
RD3	x			
RD4	x			
RD5	x			
RD6	x			
RD7	x			
RE0	x			
RE1	x			
RE2	x			

Wire Service Network Memory Map

This section defines the Wire Service Network Memory Map for the first Raptor system. Its purpose is to identify all Wire Service addressable entities and describe their function and any special information about them.

This section is incomplete yet and only a small incomplete sample is supplied. (Although some of the more complicated ones are described.)

The address format is "pp:aaaa", where "p" is the processor ID (hexadecimal) of the Wire Service Processor where the data resides and "aaaa" is the hexadecimal address or address range for the data.

Name	Type	Address	Description	Notes
WS_DESC_Pn	STRING	0n:0000	Wire Service Processor Type/Description	
WS_REV_Pn	STRING	0n:0001	Wire Service Software Revision/Date info	
WS_SB_FAN_HI	BIT	03	System Board Fans HI	Controls S1_FAN_HI. Set on 0->1 transition of WS_SB_FAN_LED. Cleared by other software
WS_SB_FAN_LED	BIT	03	System Board Fan Fault LED	Controls S1_SBFAN_LED. It is set whenever any WS_SB_FANFAULTn is set. Log 0->1 transition
WS_SB_BUSCORE	BYTE	03	System Board BUS/CORE speed ratio to use on reset	Value is asserted on S1_BC_DS[0-3] unless reading DIMM types. Set to 0 on power on.
WS_SYS_LCD	STRING	03	Value to display on LCD	For a Nx2 display the first N bytes display on top line and the second N bytes display on the bottom line. Manipulates S1_LCD_D[0-7], S1_LCD_RS, S1_LCD_ENA, S1_LCD_RW
WS_SB_FAN1	BYTE	03	System Board Fan 1 speed	Approximately every second a fan is selected by S1_FAN_SEL[0-2] and monitored via S1_FAN_TP driving a counter for a known period of time. The counter is then loaded into the appropriate fan speed. If WS_SB_FAN_HI is not set then the speed is compared against WS_SB_FAN_LOLIM. If fan is slow set appropriate WS_SB_FANFAULTn otherwise clear it.
WS_SB_FAN2	BYTE	03	System Board Fan 2 speed	
WS_SB_FAN3	BYTE	03	System Board Fan 3 speed	
WS_SB_FAN4	BYTE	03	System Board Fan 4 speed	
WS_SB_FANFAULT1	BIT	03	System Board Fan 1 Faulted	
WS_SB_FANFAULT2	BIT	03	System Board Fan 2 Faulted	

WS_SB_FANFAULT3	BIT	03	System Board Fan 3 Faulted	
WS_SB_FANFAULT4	BIT	03	System Board Fan 4 Faulted	
WS_SB_FAN_LOLIM	BYTE	03	Fan speed low speed fault limit	Set to 777 on power on
WS_SB_DIMM_SEL	BYTE	03	The DIMM select bits to use when reading DIMM_TYPE	The low order 4 bits are the select bits to use when WS_SB_DIMM_TYPE is read.
WS_SB_DIMM_TYPE	BYTE	03	The type of DIMM in the DIMM_SEL position	When read asserts value of WS_SB_DIMM_SEL on S1_BC_DS[0-3] and then returns value of S1_DIMM_D[0-7]
WS_SB_FLASH_ENA	BIT	04	Indicates FLASH ROW write enabled	Set/Cleared by debounced 0->1 transition of S2FLASH_SW. Controls state of S2_FLASH_WE and S2_FLASH_LED.
WS_SB_FRU_FAULT	BIT	04	Indicates the FRU status	At power on starts at 1. Controls S2_SBFLT_LED(0-1] for bicolor LED colors 0=Green 1=Amber, Cleared by other software
WS_SYS_OVERTEMP	BIT	04	Indicates Overtemp fault	At power on is set. Controls S2_OVRTMP_LED. Controlled by wire service backplane processor.
WS_SB_JTAG	BIT	04	Enables JTAG chain on system board	Clear at power on. Controls S2_SB_JTAG
WS_SB_CPU_PRES	BYTE	04	CPU Presence bits (LSB = CPU1)	Assemble from S2_PRES_CPU[1-4]
WS_SB_CPU_ERR	BYTE	04	CPU Error bits (LSB = CPU1)	Assemble from S2_ERROR_CPU[1-4]
WS_SB_CPU_TEMP	BYTE	04	CPU Thermal fault bits (LSB = CPU1)	Assemble from S2_TEMP_CPU[1-4]
WS_SB_CPU_POK	BYTE	04	CPU Power OK (LSB = CPU1)	Assemble from S2_POK_CPU[1-4]
WS_NMI_MASK	BYTE	04	CPU NMI processor mask (LSB=CPU1)	Defaults to all ones on power up
WS_NMI_REQ	BIT	04	NMI Request bit	When set pulse S2_NMI_CPUn corresponding to each bit set in WS_NMI_MASK. Then clear request bit. Log Action
WS_SYSFAULT	BIT	04	System Fault Summary	This bit is set if any faults detected in the system. Controls S2_SYSFLT_LED Bits scanned WS_SP_CPU_FAULT, WS_SB_FRU_FAULT (other faults?)
WS_SB_CPU_FAULT	BIT	04	CPU Fault Summary	This bit is set if ((WS_SB_CPU_ERR | WS_SB_CPU_TEMP | ~WS_SB_CPU_POK) & ~WS_SB_CPU_PRES) != 0. Log 0->1 transition with CPU bytes.

WS_BP_P5V	BYTE	02	Analog Measure of +5 volt main supply	read from S4_VOLTS_P5v
WS_BP_P3V	BYTE	02	Analog Measure of +3.3 volt main supply	read from S4_VOLTS_P3v
WS_BP_P12V	BYTE	02	Analog Measure of +12 volt main supply	read from S4_VOLTS_P12v
WS_BP_P5V	BYTE	02	Analog Measure of -12 volt main supply	read from S4_VOLTS_N12V
WS_SYS_CAN_PRES	BYTE	02	Presence bits for canisters (LSB=1, MSB=8)	controlled by S4_PSN_CAN[1-8]. A previous value byte needs to be maintained so canister transitions can be recognized. Previous value initialized to zero. Periodic monitor scans for new canisters. When new canister is recognized read full serial data and store in WS_SYS_CAN_SERIALn then log and send event
WS_SYS_PS_PRES	BYTE	02	Presence bits for power supplies (LSB=1, MSB=3)	controlled by S4_PSN_PS[1-3]. A previous value byte needs to be maintained so power supply transitions can be recognized. Previous value initialized to zero. Periodic monitor scans for new power supplies. When new power supply is recognized read full serial data and store in WS_SYS_PS_SERIALn then log and send event
WS_SYS_PS_ACOK	BYTE	02	Power supply ACOK status (LSB=1, MSB=3)	controlled by S4_ACOK_PS[1-3]. A previous value byte needs to be maintained so power supply transitions can be recognized. Previous value initialized to zero. Periodic monitor scans for changes is ACOK and sends events
WS_SYS_PS_DCOK	BYTE	02	Power supply DCOK status (LSB=1, MSB=3)	controlled by S4_DCOK_PS[1-3]. A previous value byte needs to be maintained so power supply transitions can be recognized. Previous value initialized to zero. Periodic monitor scans for changes is ACOK and sends events
WS_SYS_BP_TYPE	BYTE	02	Type of system backplane currently only two types Type 0 = 4 canister (small) and Type 1 = 8 canister (large)	controlled by S4_BP_TYPE
WS_SYS_TEMP_SB1	BYTE	02	Temperature of system board position 1	controlled by reading Dallas temperature transducers connected to serial bus on S4_TEMP_SDA and S4_TEMP_SCL
WS_SYS_TEMP_SB2	BYTE	02	Temperature of system board position 2	
WS_SYS_TEMP_BP1	BYTE	02	Temperature of backplane position 1	
WS_SYS_TEMP_BP2	BYTE	02	Temperature of backplane position 2	

WS_SYS_TEMP_WARN	BYTE	02	Warning temperature. Initialized to ???	If any WS_SYS_TEMP_ion exceeds this value, log, send event, set WS_SYS_OVERTEMP
WS_SYS_TEMP_SHUT	BYTE	02	Shutdown temperature. Initialized to ???	If any WS_SYS_TEMP_ion exceeds this value, log and clear WS_SYS_POWER
WS_SYS_REQ_POWER	BIT	02	Set to request main power on	
WS_SYS_POWER	BIT	02	Controls system master power S4_POWER_ON	When this bit is set 0->1 set S4_POWER_ON. WS_SYS_RUN = 0, WS_SYS_RSTIMER = 5 and log. When this bit is cleared clear S4_POWER_ON and log.
WS_SYS_RSTIMER	BYTE	02	Used to delay reset/run until power stabilized	Counts down to 0 at 10 counts per second. When 1->0 transition sets WS_SYS_RUN.
WS_SYS_RUN	BIT	02	Controls the system halt/run line S1_OK_TO_RUN.	If this bit is cleared, clear S1_OK_TO_RUN and log. If this bit is set, set S1_OK_TO_RUN and log.
WS_CAN_POWER	BIT	2x	Controls canister PCI slot power	When set then set S5_P5V_ENA[1_4],S5_P12V_ENA in that order with small (about 1 ms) delay between each, then log. When cleared then clear S5_P12V_ENA, S5_P5V_ENA[1_4] then log
WS_CAN_PCI_PRESENT	BYTE	2x	Reflects PCI card slot[1..4] presence indicator pins (MSB to LSB) 4B,4A,3B,3A,2B,2A,1B, 1A	Reflects data from S5_PRSNT_S[1_4][A/B]
WS_CAN_S5_PRESENT	BIT	2x	Indicates the presence of something in slot 5	Reflects S5_PRSNT_S5
WS_CAN_S5_SMART	BIT	2x	Indicates something other than a passive board in slot 5	On power up attempt to read Dallas serial number chip using S5_PSN_S5. If present set this bit and read full serial data and store in WS_SYS_CAN_IOP_SERIALn
WS_CAN_FAN_HI	BIT	2x	Canister Fans Hi	Controls S1_FAN_HI. Set on 0->1 transition of WS_SB_FAN_LED. Cleared by other software
WS_CAN_FAN_LED	BIT	2x	Canister Fan Fault LED	Controls S5_CANFAN_LED. It is set whenever any WS_CAN_FANFAULTn is set. Log 0->1 transition
WS_CAN_FANFAULT1	BIT	03	Canister Fan 1 Faulted	
WS_CAN_FANFAULT2	BIT	03	Canister Fan 2 Faulted	

WS_CAN_FAN1	BYTE	2x	Canister Fan 1 speed	Approximately every second a fan is selected by S5_FAN_SEL0 and monitored via S5_FAN_TP driving a counter for a known period of time. The counter is then loaded into the appropriate fan speed. If WS_CAN_FAN_HI is not set then the speed is compared against WS_CAN_FAN_LOLIM. If fan is slow set appropriate WS_CAN_FANFAULTn otherwise clear it
WS_CAN_FAN2	BYTE	2x	Canister Fan 2 speed	
WS_CAN_FAN_LOLI M	BYTE	2x	Fan low speed fault limit	Set to equivalent of xxx RPM on power on
WS_CAN_JTAG_ENA	BIT	2x	Enable JTAG TMS chain for canister	Copy set value to S5_CAN_JTAG
WS_CAN_NMI_S5	BIT	2x	NMI card in slot 5	when set, pulse S2_NMI_S5
WS_RI_CD	BIT	11	Status of Remote Port Modem CD	Follows S6_MODEM_CD
WS_RI_DTR	BIT	11	State of Remote Port Modem DTR	Controls S6_MODEM_DTR
WS_RI_DSR	BIT	11	Status of Remote Port Modem DSR	Follows S6_MODEM_DSR
WS_RI_RTS	BIT	11	Status of Remote Port Modem RTS	Controls S6_MODEM_RTS
WS_RI_CTS	BIT	11	Status of Remote Port Modem CTS	Follows S6_MODEM_CTS
WS_RI_CALLOUT	BYTE	11	Controls Call out Script activation	If written to it initiates Call out sequence programmed in WS_SYS_CALL_SCRIPT passing value as argument to script. Log it (Format of Script Programs TBD)
WS_RI_EVENTS	EVENT	11	Remote Interface Event Queue	See Event Data type description in prior section.
WS_SI_EVENTS	EVENT	10	System Interface Event Queue	See Event Data type description in prior section.
WS_SYS_LOG	LOG	01	System Log	The system log kept in NVRAM (See LOG data type in previous section)
WS_SYS_SCREEN	SCREE N	01	System Screen	A copy of the most recent character mode screen from the system video display (See SCREEN data type in previous section)
WS_SYS_SB_SERIAL	STRING	01	Last known System Board serial data	
WS_SYS_BP_SERIAL	STRING	01	Last known Back Plane serial data	
WS_SYS_RI_SERIAL	STRING	01	Last known Remote Interface serial data	

WS_SYS_CAN_SERIAL[1-8]	STRING	01	Last known Canister [1-8] Serial data	May be zero length if no canister ever seen
WS_SYS_IOP_SERIAL[1-8]	STRING	01	Last known IOP in Canister [1-8] Serial data	May be zero length if no canister ever seen or current canister has no IOP
WS_SI_QUEUE	QUEUE	01	Queue of data going to System Interface	See Queue data type in previous section
WS_RI_QUEUE	QUEUE	01	Queue of data going to Remote Interface	See Queue data type in previous section
WS_SYS_XDATA	BYTE ARRAY	01	Byte Array for storage of arbitrary external data in NVRAM	Wire Service just maintains this data area and is unaware of the meaning of any data stored in it.
WS_SYS_EXT_KB	BYTE	01	Size of the WS_SYS_XDATA in kilobytes	Necessary for memory management of the data area